Machine Learning

1. If your model has 100% accuracy on the training set, what should be your immediate suspicion?

Answer:

Achieving 100% accuracy on the training set is a red flag and typically indicates **overfitting**. Overfitting means the model has memorized the training data—including noise or outliers—instead of learning generalizable patterns. This leads to **poor performance on unseen data** (test or real-world).

In interview terms:

"My immediate suspicion would be that the model is over-complex or that there's a data leakage issue where future information is inadvertently being used during training."

2. Why does scaling matter more in algorithms like KNN and SVM compared to decision trees?

Answer:

Algorithms like **KNN** and **SVM** are **distance-based** or **geometric** methods. Scaling ensures that all features contribute equally to distance calculations or margin optimization.

- KNN uses Euclidean distance → unscaled features distort neighbor calculations.
- **SVM** finds the optimal hyperplane → unscaled features skew the margin.

In contrast, **Decision Trees** split data based on feature thresholds, and are **invariant to feature** scaling.

In interview terms:

"Scaling impacts models that rely on geometric concepts. Without scaling, features with larger ranges can dominate distance or margin calculations, leading to biased or incorrect results."

3. How can you decide whether your model is underfitting or overfitting just by looking at performance metrics?

Answer:

Compare training vs. validation/test scores:

Scenario	Training Accuracy	Validation Accuracy	Interpretation
Underfitting	Low	Low	Model is too simple or not trained enough
Overfitting	High	Low	Model is too complex and memorizing training data
Good Fit	High	High	Model generalizes well

In interview terms:

"If both accuracies are low, it's underfitting. If training accuracy is high but test accuracy drops significantly, it's overfitting."

4. In what situation can adding more data hurt model performance?

Answer:

Adding more data can hurt when:

- The new data is noisy or mislabeled, introducing confusion.
- The data distribution shifts (concept drift) the new data differs significantly from the training data distribution.
- Class imbalance increases, making the model biased.
- Model is under capacity it cannot handle large data well without retraining or proper tuning.

In interview terms:

"More data isn't always better. If the added data is noisy, imbalanced, or from a different distribution, it can degrade the model's performance instead of improving it."

5. What's the difference between bagging and boosting — not just technically but in terms of error handling?

Answer:

Aspect	Bagging	Boosting
Approach	Parallel training	Sequential training
Focus	Reducing variance	Reducing bias
Error Handling	Handles overfitting by averaging multiple weak learners	Focuses on correcting previous errors
Examples	Random Forest	AdaBoost, XGBoost

In interview terms:

"Bagging reduces variance by training multiple models on random subsets and averaging results. Boosting improves accuracy by focusing on mistakes made by previous models, thus reducing bias."

6. How does dropout help in preventing overfitting in deep learning models?

Answer:

Dropout randomly disables a subset of neurons during training. This prevents the network from becoming overly reliant on specific neurons and forces it to learn **redundant**, **robust representations**.

It acts like **bagging inside a neural network**, effectively training an ensemble of subnetworks.

In interview terms:

"Dropout introduces noise during training, forcing the network to generalize better. It's like training different subnetworks every iteration, which helps prevent overfitting."

7. Why is model interpretability crucial in domains like healthcare and finance?

Answer:

These domains involve **high-stakes decisions**:

- Healthcare: Predicting diseases or treatments needs explainability to ensure trust and accountability.
- **Finance**: Credit scoring, fraud detection, or trading decisions require transparency for **compliance** and **regulatory approval**.

In interview terms:

"In critical domains, stakeholders must understand and trust the model's decision-making process. Black-box models can pose legal, ethical, and safety risks."

8. Why does regularization help improve generalization?

Answer:

Regularization (like **L1** or **L2**) adds a penalty to large weights, discouraging complex models.

- L1 leads to **sparsity** → feature selection
- L2 leads to weight shrinkage → less complex models

By constraining the model's flexibility, regularization reduces overfitting and improves performance on unseen data.

In interview terms:

"Regularization controls model complexity, making it more robust and less likely to memorize training data noise."

9. When is it better to use a shallow model over a deep one?

Answer:

Use **shallow models** when:

- Data is limited → Deep models overfit.
- Model interpretability is important
- Problem is linearly separable or low-dimensional
- Training time or resources are constrained

"If the data is small, the problem is simple, or interpretability matters, shallow models are more practical and performant than deep networks."

10. What are some real-world examples where linear models outperform deep learning models?

Answer:

- **Credit Scoring** (Logistic Regression): Highly regulated, interpretable.
- **Click-through rate prediction** in online advertising (e.g., Logistic Regression with large-scale features).
- Spam detection
- Medical risk scoring
- A/B Testing Models (linear regression or logistic regression)

In interview terms:

"In domains with tabular data, low complexity, and high interpretability needs—like banking or marketing—linear models often outperform deep learning due to simplicity, speed, and transparency."

Let me know if you'd like these turned into a PDF, LinkedIn post, or flashcards for revision.

You said:

#Natural_Language_Processing_(NLP)_Questions 1. What is the difference between context-free and context-aware embeddings? 2. How does self-attention allow transformers to

scale better than RNNs for long sequences? 3. Why is sentence segmentation more challenging in multilingual or informal texts? 4. What are hallucinations in large language models, and why do they occur? 5. How can prompt engineering improve NLP outputs without changing the model? 6. How does zero-shot learning work in NLP models?

hashtag#Statistics Probability Questions 1. Why can a statistically significant result still be practically insignificant? 2. What assumptions must be satisfied for linear regression to work well? 3. What's the role of degrees of freedom in statistical tests? 4. How can outliers affect the mean and standard deviation? 5. What's the difference between descriptive and inferential statistics, and where do we use each? hashtag#Classification Questions 1. In a multi-class classification problem, why is micro vs. macro averaging important? 2. How does SMOTE work, and what are its limitations? 3. What causes a model to have high recall but low precision? 4. Why might your model predict only one class in an imbalanced dataset? 5. How does label encoding differ from one-hot encoding in the context of classification?

ChatGPT said:

Updated saved memory

Here's a **detailed interview-ready explanation** for each question across the three categories: NLP, Statistics & Probability, and Classification.



#Natural_Language_Processing_(NLP)_Questions

1. What is the difference between context-free and context-aware embeddings?

Context-Free Embeddings (e.g., Word2Vec, GloVe):

- Each word has a single vector representation, regardless of the sentence it appears in.
- "Bank" has the same vector in both "river bank" and "bank account".

Context-Aware Embeddings (e.g., BERT, GPT):

- Word representation depends on the surrounding words (context).
- "Bank" has different embeddings depending on usage.

In interview terms:

"Context-aware embeddings capture semantics better by considering the word's role in a sentence, improving downstream NLP task performance."

2. How does self-attention allow transformers to scale better than RNNs for long sequences?

Self-Attention (Transformers):

- Processes all tokens in parallel.
- Learns relationships between all pairs of words, regardless of distance.
- Time complexity: O(n²), but fully parallelizable.

RNNs:

- Sequential processing → **slow**, can't parallelize.
- Long-term dependencies suffer due to vanishing gradients.

In interview terms:

"Self-attention lets the model directly relate distant words without step-by-step recurrence, enabling faster training and better long-range understanding."

3. Why is sentence segmentation more challenging in multilingual or informal texts?

- Multilingual texts may not follow punctuation norms.
- Informal texts (e.g., social media) lack standard grammar.
- Emojis, abbreviations, slang, and missing punctuation confuse traditional rule-based or ML-based sentence boundary detectors.

In interview terms:

"Language variation and informality break conventional rules, making sentence boundary detection harder for multilingual or casual language."

4. What are hallucinations in large language models, and why do they occur?

Hallucinations are confident but factually incorrect outputs generated by LLMs.

Why they occur:

- LLMs optimize for **plausibility**, not factual accuracy.
- Lack grounding to external knowledge bases unless explicitly trained.
- Often generate based on **statistical patterns**, not truth.

In interview terms:

"Hallucinations happen when models generate fluent but untrue content because they rely on patterns, not verified facts."

5. How can prompt engineering improve NLP outputs without changing the model?

- Carefully designing input prompts guides the model's behavior.
- Techniques: few-shot, zero-shot, chain-of-thought, role instructions.
- Useful in tasks like summarization, reasoning, or translation.

In interview terms:

"Prompt engineering steers pre-trained models by crafting inputs strategically, often improving output quality without retraining the model."

6. How does zero-shot learning work in NLP models?

- Model performs tasks without being trained on task-specific data.
- Relies on pretraining with instructions or task descriptions.
- Example: Ask GPT to summarize a text even if no summarization data was used during fine-tuning.

"Zero-shot learning leverages language understanding learned during pretraining to generalize to new tasks, guided only by natural language prompts."

#Statistics_Probability_Questions

1. Why can a statistically significant result still be practically insignificant?

- Statistical significance: Result unlikely due to chance (p < 0.05).
- Practical significance: Real-world impact is meaningful.
- Large sample sizes can detect tiny effects that are statistically significant but irrelevant in practice.

In interview terms:

"A statistically significant result may have no practical value if the effect size is too small to matter in real-world applications."

2. What assumptions must be satisfied for linear regression to work well?

- 1. **Linearity** between predictors and response.
- 2. **Independence** of observations.
- 3. Homoscedasticity (constant variance of errors).

- 4. Normality of residuals.
- 5. No **multicollinearity** among predictors.

"Violating these assumptions can lead to biased, inefficient, or misleading estimates."

3. What's the role of degrees of freedom in statistical tests?

- Represents the number of values in the calculation that are free to vary.
- Affects **critical values** for t-tests, chi-square tests, etc.
- More degrees of freedom → more reliable estimates.

In interview terms:

"Degrees of freedom adjust for model complexity and sample size, ensuring accurate confidence intervals and hypothesis tests."

4. How can outliers affect the mean and standard deviation?

- Mean is sensitive to extreme values → gets pulled toward outliers.
- Standard deviation increases → misrepresents spread.
- Median and IQR are more robust in such cases.

In interview terms:

"Outliers can distort central tendency and variability, leading to misleading conclusions in analysis."

5. What's the difference between descriptive and inferential statistics, and where do we use each?

Type	Purpose	Example
Descriptive	Summarize data	Mean, median, standard deviation
Inferential	Draw conclusions about population	Hypothesis testing, confidence intervals

"Descriptive stats describe what we see. Inferential stats help us generalize from samples to populations."



1. In a multi-class classification problem, why is micro vs. macro averaging important?

- Macro Average: Treats all classes equally, even if imbalanced.
- Micro Average: Aggregates all outcomes, favoring frequent classes.

In interview terms:

"Macro is good for evaluating model performance across all classes equally. Micro is better when class imbalance is present."

2. How does SMOTE work, and what are its limitations?

• **SMOTE** (Synthetic Minority Oversampling Technique): Creates synthetic samples by interpolating between minority class neighbors.

Limitations:

- Can create ambiguous or noisy samples.
- Doesn't address majority class dominance.

Doesn't reduce overlap between classes.

In interview terms:

"SMOTE helps with class imbalance, but can introduce unrealistic points or worsen overlapping between classes."

3. What causes a model to have high recall but low precision?

- Model is **lenient** in predicting positives.
- Captures most true positives (high recall) but includes many false positives (low precision).

Common in: Fraud detection, disease screening (where missing a positive is worse).

In interview terms:

"The model prefers catching all positives, even at the cost of wrongly labeling negatives as positive."

4. Why might your model predict only one class in an imbalanced dataset?

- Bias toward majority class: Model minimizes loss by predicting the dominant class.
- No penalty for missing minority class.
- Poor representation of minority class in training.

In interview terms:

"If data is imbalanced, the model may optimize for accuracy by ignoring minority classes. Techniques like class weights or resampling are needed."

5. How does label encoding differ from one-hot encoding in the context of classification?

- Label Encoding: Converts categories to integers. Implies ordinal relationship.
- One-Hot Encoding: Creates binary columns for each category. No ordinal meaning.

Use one-hot when categories are nominal (no order), and label encoding when categories are ordinal or for tree-based models that handle it well.

In interview terms:

"Label encoding can mislead algorithms into thinking one category is higher than another. One-hot avoids this by creating independent binary features."

Let me know if you'd like a PDF summary, cheatsheet, or a mock interview-style Q&A!

You said:

1/ Linear Models: • Linear Regression: Linearity, independence, homoscedasticity, normality, Ordinary Least Squares, cost function, gradient descent. • Logistic Regression: Sigmoid function, cross-entropy loss, decision boundaries, probability interpretation. 2/ Tree-Based Algorithms: • Decision Trees: How they work (splitting criteria like Gini impurity, entropy), advantages (interpretability), disadvantages (overfitting). • Random Forests: Ensemble method (bagging), reducing variance, feature importance. 3/ Support Vector Machines (SVMs): • Concept: Maximizing the margin, support vectors. • Kernels: Linear, Polynomial, RBF (understanding the kernel trick for non-linear separation). • Regularization (C parameter): Handling misclassifications vs. wide margins. 4/ K-Nearest Neighbors (k-NN): • Concept: Instance-based learning, lazy learning. • Distance metrics: Euclidean, Manhattan. • Curse of Dimensionality: Its impact on k-NN. 5/ Naive Bayes: • Concept: Probabilistic classifier based on Bayes' Theorem with a "naive" independence assumption. • Types: Gaussian, Multinomial, Bernoulli. 6/ Clustering Algorithms (Unsupervised Learning): • K-Means Clustering: How it works (centroid calculation, iteration), choosing k (elbow method, silhouette score). • Hierarchical Clustering: Agglomerative vs. Divisive, dendrograms. 7/ Dimensionality Reduction: • Principal Component Analysis (PCA): Concept (eigenvectors, eigenvalues), variance explained, applications (feature extraction, visualization). 8/Ensemble Methods (Beyond Trees): • Bagging: Bootstrap aggregating (e.g., Random Forests). • Boosting: Sequentially building models (e.g., AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost – know their differences and why they're powerful). Crucial Concepts to Discuss for ANY Algorithm: • Assumptions: What does the algorithm assume about the data? • Strengths & Weaknesses: When is it a good choice. and when is it not? • Hyperparameters: Key parameters and their impact on model performance. • Bias-Variance Trade-off: How each algorithm balances this. • Overfitting & Underfitting: How to diagnose and mitigate. • Feature Scaling: When and why it's necessary. • Computational Complexity: Big O notation for training and prediction.

ChatGPT said:

Updated saved memory

Great — you're compiling a comprehensive set of interview topics. Here's a **detailed breakdown** of each algorithm and concept with an **interview-oriented perspective**, keeping answers crisp yet thorough for explanation during interviews:

1. Linear Models

Linear Regression

- **Assumptions**: Linearity, independence, homoscedasticity, normality of residuals.
- OLS (Ordinary Least Squares): Minimizes the sum of squared residuals.
- Cost Function: MSE = $(1/n) \Sigma (y_i \hat{y}_i)^2$
- Optimization: Gradient descent (especially in large datasets).
- **Strengths**: Simple, interpretable.
- Weaknesses: Poor with multicollinearity, outliers, and non-linearity.
- Scaling: Required for gradient descent, not OLS directly.
- **Complexity**: O(n²) training (matrix inversion).

Logistic Regression

- **Sigmoid Function**: Converts linear output to probability [0, 1].
- Loss Function: Binary cross-entropy (log loss).
- Interpretation: Coefficients = log odds.
- **Decision Boundary**: Where predicted probability = 0.5.
- Regularization: L1 (Lasso) for feature selection, L2 (Ridge) for shrinkage.
- **Scaling**: Important for convergence when using gradient-based solvers.

2. Tree-Based Algorithms

Decision Trees

- How it works: Greedy splitting using Gini Impurity or Entropy.
- Advantages: No scaling needed, interpretable.
- **Disadvantages**: Prone to overfitting without pruning.
- **Complexity**: O(n log n) for training (best split computation).

Random Forest

- Concept: Bagging with multiple decision trees.
- Reduces Variance: Averaging decorrelates errors.
- Feature Importance: Based on decrease in impurity.
- Hyperparameters: n estimators, max depth, max features.
- **Strength**: Handles non-linearity, robust to overfitting.
- Weakness: Less interpretable than a single tree.

3. Support Vector Machines (SVMs)

- **Concept**: Find the hyperplane that maximizes margin between classes.
- **Support Vectors**: Critical points near the boundary.
- C Parameter:
 - \circ High C \rightarrow Low bias, overfitting risk.
 - \circ Low C \rightarrow High bias, more generalization.

Kernels:

- Linear: Simple linear separation.
- Polynomial: Degree-based curves.
- **RBF**: Maps data into high-dimensional space.
- Kernel Trick: Computes dot products in higher space efficiently.

4. K-Nearest Neighbors (k-NN)

- Concept: Lazy learner, classifies based on majority vote of neighbors.
- **Distance Metrics**: Euclidean, Manhattan, Minkowski.
- Curse of Dimensionality: Distances become less meaningful in high dimensions.
- Scaling: Mandatory for distance computation.
- **Hyperparameter**: k lower k may overfit, higher k may underfit.
- Complexity:
 - o Training: O(1)
 - Prediction: O(n) per query (can be slow).

5. Naive Bayes

- **Concept**: Based on Bayes' theorem assuming feature independence.
- Formula:
 - $P(y|X)=P(X|y)P(y)P(X)P(y|X) = \frac{P(X|y)P(y)}{P(X)}P(y|X)=P(X)P(X|y)P(y)$
- Types:

- o Gaussian: For continuous data (assumes normal distribution).
- Multinomial: Count data (e.g., text classification).
- Bernoulli: Binary features.
- Strengths: Simple, fast, works well for text data.
- Weaknesses: Strong independence assumption.

6. Clustering Algorithms (Unsupervised)

K-Means

- **Concept**: Assigns points to k centroids, minimizes within-cluster variance.
- Initialization: Random or KMeans++ (better convergence).
- Choosing K: Elbow method, silhouette score.
- **Limitation**: Sensitive to initialization, assumes spherical clusters.

Hierarchical Clustering

- Agglomerative: Bottom-up (merge closest).
- **Divisive**: Top-down (split recursively).
- **Dendrogram**: Tree diagram showing cluster merging.
- Strength: No need to specify k in advance.
- Weakness: Computationally expensive.

7. Dimensionality Reduction

PCA (Principal Component Analysis)

- **Concept**: Projects data to directions of maximum variance.
- **Eigenvectors**: New axes.
- Eigenvalues: Variance explained.
- **Applications**: Feature reduction, noise filtering, visualization.
- **Limitation**: Linear only; sensitive to scaling.

8. Ensemble Methods

Bagging

- Concept: Training on different bootstrap samples, aggregate predictions.
- Example: Random Forest.
- Strength: Reduces variance.

Boosting

- **Concept**: Sequential learning, each model fixes previous errors.
- Examples:
 - AdaBoost: Weights updated based on previous errors.
 - Gradient Boosting: Optimizes loss using gradient descent.
 - **XGBoost**: Regularized version, fast & efficient.
 - LightGBM: Leaf-wise growth, faster on large data.
 - CatBoost: Handles categorical features natively.
- Strength: High performance, handles bias.

• Weakness: Overfitting risk, complex tuning.

Crucial Concepts for ANY Algorithm

Concept Description

Assumptions What the model expects about data (e.g., normality in linear

regression)

Strengths/Weaknesses Understand model fit and failure cases

Hyperparameters Control complexity (e.g., k in k-NN, max_depth in trees)

Bias-Variance Trade-off Bias = underfitting, variance = overfitting

Overfitting/Underfitting Diagnosed via train-test gap

Feature Scaling Needed for gradient-based and distance-based models

Computational Time/memory costs — important for production or large-scale

Complexity data

Would you like:

- A printable summary/PDF?
- A cheat sheet with formulas + tips?
- A set of interview mock questions based on this?

Let me know how you'd like to use this material!

You said:

1. Why can a high training accuracy be misleading when evaluating a model's performance? 2. What's the intuition behind the bias-variance trade-off? 3. Why is random initialization crucial in training neural networks? 4. In ensemble methods like random forests, why is averaging predictions better than taking a majority vote? 5. How does L1 regularization differ from L2, and when would you use each? 6. What's the purpose of using an activation function in neural networks? 7. How does increasing the number of estimators in a random forest impact model performance? 8. Why can using too many features lead to worse model performance, and how can you detect it? 9. How does early stopping work in deep learning, and why is it useful? 10.

What is data leakage, and how can it affect your machine learning model? #Natural_Language_Processing_(NLP)_Questions 1. Why is stemming sometimes less accurate than lemmatization? 2. How does positional encoding help transformer-based models understand word order? 3. What are the challenges in handling code-mixed languages (e.g., Hinglish) in NLP? 4. Why are subword tokenization techniques (like Byte-Pair Encoding) preferred in large language models? 5. What is the vanishing gradient problem, and how do RNNs suffer from it? 6. Why is fine-tuning pre-trained models more effective than training from scratch in NLP? #Statistics Probability Questions 1. What is the central limit theorem, and why is it important in data science? 2. When would you prefer a non-parametric test over a parametric one? 3. Why does increasing sample size reduce sampling error but not necessarily eliminate bias? 4. How do confidence intervals help interpret statistical results? 5. What is the difference between Type I and Type II errors, and how do they affect model decisions? #Classification Questions 1. What is a confusion matrix, and how does it help evaluate model performance? 2. Why is ROC-AUC a better evaluation metric in imbalanced classification problems? 3. How can class imbalance be addressed during training? 4. What's the role of regularization in logistic regression? 5. How does a decision boundary change when using different kernels in SVM?

ChatGPT said:

Updated saved memory

Excellent — you've assembled another well-rounded and essential set of interview questions. Here's a **detailed**, **interview-optimized answer** for each question across the categories:



General ML Questions

1. Why can a high training accuracy be misleading when evaluating a model's performance?

It may indicate **overfitting** — the model learns patterns and noise specific to the training data but fails to generalize. Real performance should be assessed using validation/test accuracy.

2. What's the intuition behind the bias-variance trade-off?

Bias: Error due to simplifying assumptions \rightarrow leads to underfitting. **Variance**: Error due to sensitivity to training data \rightarrow leads to overfitting. The goal is to find a balance where both are minimized for optimal generalization.

3. Why is random initialization crucial in training neural networks?

It prevents **symmetry** — if weights are initialized the same, neurons learn the same thing. Randomization ensures diverse feature learning. Poor initialization can also slow or prevent convergence.

4. In ensemble methods like random forests, why is averaging predictions better than taking a majority vote?

For **regression**, averaging reduces variance and gives smoother output. For **classification**, majority voting is common, but **probability averaging** (soft voting) can often outperform hard voting by incorporating confidence.

5. How does L1 regularization differ from L2, and when would you use each?

L1 (Lasso) L2 (Ridge)

Adds w

Encourages sparsity Shrinks weights

Useful for feature selection Useful for multicollinearity

Use L1 when feature selection is important, L2 when you want to keep all features but control their influence.

6. What's the purpose of using an activation function in neural networks?

Without activation functions, neural networks are just linear models. Activation functions introduce **non-linearity**, enabling the network to model complex functions.

7. How does increasing the number of estimators in a random forest impact model performance?

Initially, it improves performance by reducing **variance**, but after a point, returns diminish. It also increases **training time** and **memory use**.

8. Why can using too many features lead to worse model performance, and how can you detect it?

This leads to the **curse of dimensionality**:

- Increases overfitting risk
- Redundant/noisy features degrade performance

Use feature importance, PCA, or regularization to detect and manage it.

9. How does early stopping work in deep learning, and why is it useful?

It monitors validation loss. Training halts if loss doesn't improve for 'n' epochs, preventing overfitting and saving time.

10. What is data leakage, and how can it affect your machine learning model?

Occurs when **information from outside the training set** leaks into the model, often during preprocessing (e.g., scaling on entire dataset). This leads to **inflated accuracy** and poor generalization.

#Natural_Language_Processing_(NLP)_Questions

1. Why is stemming sometimes less accurate than lemmatization?

Stemming chops off word endings without context ("running" \rightarrow "run"), possibly producing **non-words**. Lemmatization uses **morphological analysis** to return meaningful base forms.

2. How does positional encoding help transformer-based models understand word order?

Transformers lack recurrence. Positional encoding adds **sequence information** (e.g., sin/cos functions) to embeddings, helping the model distinguish between "cat sat on mat" vs. "mat sat on cat".

3. What are the challenges in handling code-mixed languages (e.g., Hinglish) in NLP?

- Lack of standardized grammar/spelling
- Multilingual vocabulary
- Ambiguity in tokenization and translation

These pose challenges for **tokenization**, **language identification**, and **model training**.

4. Why are subword tokenization techniques (like Byte-Pair Encoding) preferred in large language models?

They:

- Handle rare/unknown words
- Reduce vocabulary size
- Preserve semantic units (e.g., "unhappiness" → "un", "happi", "ness")

This improves generalization and memory efficiency.

5. What is the vanishing gradient problem, and how do RNNs suffer from it?

During backpropagation, gradients shrink through time steps \rightarrow weights stop updating. RNNs struggle with **long-term dependencies** due to this. LSTMs/GRUs were designed to address it.

6. Why is fine-tuning pre-trained models more effective than training from scratch in NLP?

Pre-trained models (like BERT) already understand grammar, semantics, etc. Fine-tuning just adapts them to the task, saving time and data. Training from scratch is expensive and rarely better.



#Statistics_Probability_Questions

1. What is the central limit theorem, and why is it important in data science?

CLT states: The sampling distribution of the sample mean approaches a **normal distribution** as sample size grows, regardless of population distribution.

Enables use of parametric tests, confidence intervals, and hypothesis testing.

2. When would you prefer a non-parametric test over a parametric one?

When:

- Data is not normally distributed
- Sample size is small
- Outliers are present

Examples: Wilcoxon, Mann-Whitney tests.

3. Why does increasing sample size reduce sampling error but not necessarily eliminate bias?

Larger samples give better estimates (reduce **random error**) but **systematic bias** remains unless the data collection process is fixed (e.g., poor survey design).

4. How do confidence intervals help interpret statistical results?

A 95% CI means: If we repeated the experiment many times, 95% of the intervals would contain the true population parameter.

It provides a range estimate with uncertainty quantification.

5. What is the difference between Type I and Type II errors, and how do they affect model decisions?

Type	Meaning	Consequence
Type I	False Positive	Rejecting a true null
Type II	False Negative	Failing to reject a false null

Type I is controlled via alpha (significance level), and Type II via power (1 - beta).

#Classification_Questions

1. What is a confusion matrix, and how does it help evaluate model performance?

A confusion matrix shows TP, FP, TN, FN. It helps derive metrics like:

- **Precision** = TP / (TP + FP)
- Recall = TP / (TP + FN)

• **F1-score** = Harmonic mean of precision and recall

It gives a **complete picture** beyond accuracy.

2. Why is ROC-AUC a better evaluation metric in imbalanced classification problems?

ROC-AUC considers **true/false positive rates** across thresholds. It's threshold-independent and less biased by class imbalance compared to accuracy.

3. How can class imbalance be addressed during training?

- Resampling (oversampling minority, undersampling majority)
- SMOTE
- Class weights
- Ensemble methods like Balanced Random Forest

4. What's the role of regularization in logistic regression?

Prevents **overfitting** by penalizing large weights:

- **L1** → Feature selection
- **L2** → Shrinks all weights

Improves generalization on unseen data.

5. How does a decision boundary change when using different kernels in SVM?

- Linear Kernel: Straight line/plane
- Polynomial/RBF: Curved or complex boundaries

Kernels allow the model to separate data that's **not linearly separable** in original space.

Would you like a PDF guide, flashcards, or mock interview prep based on these?

You said:

#Feature Engineering Questions 1. Your model performs well during training but poorly on test data. Could feature leakage be the reason? ? *Hint:* If future information accidentally leaks into training features, it leads to over-optimistic performance. Audit features carefully. 2. You suspect multicollinearity in your regression model. How would you detect and resolve it? 💡 *Hint:* Use Variance Inflation Factor (VIF) to identify highly correlated predictors and consider dimensionality reduction like PCA. 3. You're working with categorical variables that have high cardinality. What's your encoding strategy? ? *Hint:* Use target encoding, frequency encoding, or embeddings instead of one-hot to avoid high dimensionality. #Time Series Questions 4. Your time series model performs poorly despite good cross-validation scores. What's likely wrong? 🂡 *Hint:* You might be using random CV splits — switch to time-based cross-validation to preserve temporal order. 5. Your ARIMA model's residuals are autocorrelated. What does this suggest? ? *Hint:* The model isn't capturing all temporal structure — try adding seasonal terms or moving to SARIMA/Prophet. 6. You're forecasting demand, but COVID-19 caused a huge spike in 2020. How do you handle such anomalies? 💡 *Hint:* Treat them as outliers either remove, smooth, or model them separately as shocks or interventions. #ML Deployment / Production Questions 7. Your deployed model suddenly starts making worse predictions. What's the first thing to check? 💡 *Hint:* Check for data drift or concept drift — retrain the model or add monitoring pipelines. 8. Your model is slow during inference in production. What could be optimized? 💡 *Hint:* Quantization, model pruning, using a faster algorithm, or switching to a lightweight framework (ONNX, TensorRT). 9. You notice discrepancies between training and production predictions. What's going wrong? 💡 *Hint:* Check for training-serving skew inconsistent preprocessing, version mismatches, or feature changes. #Bonus: General Tricky 10. You build a perfect model, but business stakeholders are still unsatisfied. Why? 💡 *Hint:* Model performance ≠ business impact — focus on interpretability, trust, and aligning with KPIs. 11. A colleague says adding more features always improves model performance. Agree? 💡 *Hint:* Not necessarily more features can lead to overfitting, multicollinearity, or noise. 12. You're asked to choose between a slightly less accurate but interpretable model vs. a black-box model. What factors guide your choice? \(\frac{1}{2} \) *Hint:* Consider domain requirements, risk, regulatory needs, and explainability trade-offs.

ChatGPT said:

Updated saved memory

Great! Here's a **concise**, **interview-optimized explanation** for each question in your new set, including **feature engineering**, **time series**, **deployment**, and **business alignment** topics:



Feature Engineering Questions

1. Could feature leakage be the reason for poor test performance despite high training accuracy?

Yes. Feature leakage occurs when information from the future or test set is used during training. It leads to **over-optimistic training performance** and **poor generalization**.

Fix: Carefully audit features and ensure proper temporal and logical separation.

2. How would you detect and resolve multicollinearity in a regression model?

Use VIF (Variance Inflation Factor):

• VIF > 5 or 10 → multicollinearity concern.

% Fix:

- Drop correlated features.
- Combine them.
- Use **PCA** or **regularization** (Ridge handles it well).

3. How do you encode high-cardinality categorical variables?

Avoid one-hot — it leads to **sparse**, **high-dimensional data**.

X Alternatives:

- **Target encoding** (with smoothing to prevent leakage)
- Frequency encoding
- **Embeddings** (for deep learning models)

Time Series Questions

4. Why might a time series model underperform despite good CV scores?

Likely due to using **random CV**, which breaks temporal structure.

Fix: Use **time-based CV** (e.g., expanding window, rolling forecast) to maintain chronological order.

5. Your ARIMA residuals are autocorrelated. What does this mean?

The model hasn't captured all temporal dependencies.

X Fix:

- Try **SARIMA** (adds seasonality)
- Check ACF/PACF plots.
- Consider ARIMA + exogenous variables (ARIMAX) or Prophet.

6. How do you handle COVID-19-like anomalies in time series forecasting?

Such events are outliers or shocks.



• Treat as outliers (remove or smooth).

- Add intervention variables or dummy indicators.
- Use robust models that can handle structural breaks.



ML Deployment / Production Questions

7. What's the first thing to check when a deployed model degrades in performance?

Check for data drift (input distribution changes) or concept drift (target concept shifts).

Fix: Monitor inputs & predictions. Retrain the model with fresh data.

8. What can be optimized if your model is slow during inference?

Speed issues can be due to:

- Model size
- Inefficient architecture
- Incompatible runtime



- Model quantization (lower precision)
- **Pruning** (remove unnecessary neurons)
- Use **ONNX**, **TensorRT**, or lighter models (e.g., LightGBM instead of XGBoost).

9. Why might training and production predictions differ?

Likely due to training-serving skew.

Causes:

- Different feature engineering
- Inconsistent libraries or versions
- Data pipelines differ

Fix: Ensure reproducibility, versioning, and integrated preprocessing pipelines (e.g., sklearn.pipeline or TF Transform).

Bonus: General Tricky Questions

10. Why might stakeholders be unsatisfied with a high-performing model?

High accuracy ≠ Business value. Stakeholders care about:

- Interpretability
- Actionability
- Alignment with business KPIs
- Trust and explainability

Tip: Speak their language and connect metrics to business impact.

11. Does adding more features always improve model performance?

No. It can lead to:

- Overfitting
- Multicollinearity

• More noise than signal

X Use feature selection, regularization, or dimensionality reduction.

12. When to choose an interpretable model over a black-box model?

Depends on:

- **Domain** (healthcare, finance require explainability)
- Regulations
- Risk tolerance
- Stakeholder trust

Choose interpretable models when **transparency** is more important than slight accuracy gains.

1. What's the difference between linear and logistic regression?

- Linear Regression: Predicts a continuous value. Uses OLS (Ordinary Least Squares) to minimize the squared error between actual and predicted.
- Logistic Regression: Predicts a binary or multi-class category. Uses sigmoid function to map output between 0 and 1. Optimized with cross-entropy loss.

2. How do you interpret coefficients in linear regression?

- Each coefficient represents the expected change in the dependent variable for a one-unit increase in the predictor variable, assuming all other variables are constant.
- If a coefficient is negative, it has an inverse relationship with the target.

3. What assumptions does linear regression make?

- Linearity: Relationship between predictors and target is linear.
- **Independence**: Residuals are independent.
- Homoscedasticity: Constant variance of residuals.
- Normality: Residuals are normally distributed.
- No multicollinearity: Predictors should not be highly correlated.

4. What is regularization? Difference between L1 and L2?

- Regularization prevents overfitting by **penalizing large coefficients**.
 - L1 (Lasso): Adds absolute value of coefficients. Can shrink some coefficients to zero (useful for feature selection).
 - L2 (Ridge): Adds squared coefficients. Shrinks values smoothly, but keeps all features.

5. How do you handle multicollinearity?

- Detect with:
 - Correlation matrix
 - VIF (Variance Inflation Factor)
- Resolve by:
 - Removing correlated features
 - Applying PCA
 - Using regularization (Ridge)

6. What metrics would you use to evaluate a classification model?

- Accuracy
- Precision, Recall, F1 Score
- ROC-AUC
- Log Loss
- Use **confusion matrix** to break down predictions.

7. Explain ROC curve and AUC.

- ROC Curve: Plots True Positive Rate vs. False Positive Rate at different thresholds.
- AUC (Area Under the Curve): Probability the model ranks a random positive higher than a random negative. Higher AUC = better performance.

▼ Trees & Ensembles

8. How does a decision tree decide where to split?

- Uses impurity measures like:
 - o Gini Impurity
 - Entropy (Information Gain)
- Chooses split that results in highest reduction in impurity.

9. What's Gini vs entropy?

- Both measure impurity.
 - Gini: Faster to compute. Tends to isolate most frequent class.
 - Entropy: More informative but slightly costlier. Based on information theory.

10. Why do decision trees overfit?

- They learn **specific rules** for training data and can capture noise.
- Overfitting occurs especially when:
 - Depth is large
 - Leaves contain very few samples

11. How does Random Forest reduce overfitting?

- Uses bagging:
 - Multiple trees trained on bootstrapped data.
 - Random feature selection at each split.
- Aggregates predictions to reduce variance and overfitting.

12. How does boosting work (e.g., AdaBoost, XGBoost)?

- Boosting = **sequential learning**:
 - Each new model tries to fix errors made by previous ones.
 - Final prediction = weighted sum of weak learners.
- XGBoost adds regularization and parallelization for speed & performance.

13. Differences between bagging and boosting?

Aspect	Bagging	Boosting
Learning	Parallel	Sequential
Aim	Reduce variance	Reduce bias
Example	Random Forest	AdaBoost, XGBoost
Model type	Independent	Trees correct each
	trees	other

Model Evaluation & Validation

14. What is bias-variance trade-off?

- **Bias**: Error from wrong assumptions (underfitting).
- Variance: Error from sensitivity to small changes (overfitting).
- Goal: Find model that balances both.

15. Explain k-fold cross-validation.

- Split data into **k parts**.
- Train on k-1 parts, test on remaining.

- Repeat k times, each fold gets to be test once.
- Helps evaluate model stability and generalizability.

16. How do you handle imbalanced classes?

- Techniques:
 - SMOTE, undersampling
 - Class weights
 - F1 score, ROC-AUC instead of accuracy
 - Ensemble models with balanced boosting

17. What is precision vs recall?

- **Precision**: Out of predicted positives, how many are truly positive.
- Recall: Out of actual positives, how many were predicted correctly.
- Use **F1 Score** for balance.

Clustering & Unsupervised Learning

18. How does k-means clustering work?

- 1. Randomly choose k centroids
- 2. Assign points to nearest centroid
- 3. Recompute centroids

4. Repeat until convergence

19. How do you choose the value of k in k-means?

- Elbow method: Plot SSE vs. k and look for "elbow"
- Silhouette score: Measures cohesion vs separation

20. Explain PCA and how it helps in ML.

- PCA = dimensionality reduction.
- Projects data to new axes (principal components) that explain max variance.
- Helps with:
 - Visualization
 - Noise reduction
 - Speeding up training

21. What's the difference between PCA and LDA?

- **PCA**: Unsupervised; maximizes variance.
- LDA: Supervised; maximizes class separability.

22. When would you use hierarchical clustering?

When:

- You don't know k in advance.
- Want a dendrogram to visualize cluster hierarchy.
- Data is small or medium-sized.

Feature Engineering & Data Preparation

23. How do you handle missing data?

- Drop rows/columns (if few).
- Impute:
 - Mean/Median/Mode
 - o KNN or regression
- Use indicators for missingness.

24. What's feature scaling and why is it important?

- Standardizes feature ranges.
- Important for:
 - Distance-based algorithms (KNN, SVM)
 - Gradient-based optimization
- Common methods: MinMax, StandardScaler

25. Explain one-hot vs label encoding.

- One-hot: Creates binary columns (for nominal categorical data).
- Label encoding: Assigns integers (for ordinal categorical data).
- Use carefully: label encoding can imply order where none exists.

26. How do you detect outliers?

- IQR Method, Z-score
- Boxplots, scatter plots
- Isolation Forest, DBSCAN

General ML Knowledge

27. What is overfitting? How do you prevent it?

- Model fits training data too well, fails to generalize.
- Prevention:
 - Regularization
 - Cross-validation
 - Pruning (trees)
 - Dropout (NNs)

28. Explain the curse of dimensionality.

• As dimensions increase:

- Data becomes sparse
- Distance metrics lose meaning
- Impacts KNN, clustering
- Fix with PCA, feature selection

29. What's the difference between parametric and non-parametric models?

- **Parametric**: Fixed # of parameters (e.g., Linear Regression)
- Non-parametric: Flexible structure, grows with data (e.g., KNN, trees)

30. How do you select the right model for your data?

- Depends on:
 - Data size
 - Interpretability
 - Accuracy needs
 - Training time
 - Business constraints
- Use experimentation, cross-validation, and domain understanding

Machine Learning & Model Selection

- 1. If two models have the same accuracy, how do you decide which one to deploy?
 - Check for precision, recall, F1-score, and AUC-ROC.
 - o Consider model complexity, interpretability, inference time, and resource usage.
 - Evaluate business needs (e.g., cost of false positives).
- Your validation accuracy is higher than test accuracy what could be the issue?

- Test set may not represent the same distribution.
- Overfitting on the validation set during hyperparameter tuning.
- Possible data leakage in the validation phase.
- 3. Why might accuracy increase, but precision drop, after hyperparameter tuning?
 - The model may start favoring the majority class to boost overall accuracy.
 - Precision penalizes false positives; tuning might be sacrificing this for recall or accuracy.
- 4. Can a model with higher bias outperform a low-bias one in production? When and why?
 - Yes, if the low-bias model overfits.
 - Simpler high-bias models may generalize better to unseen data.
- 5. Why is cross-validation not enough to say a model will generalize well?
 - It assumes the data distribution stays the same.
 - Doesn't account for concept drift, adversarial cases, or unseen variations in real-world settings.
- 6. A complex ensemble gives a 1% gain in test score over a simple model would you deploy it? Why or why not?
 - o Depends on:
 - Business value of that 1% gain.
 - Model latency, interpretability, maintenance.
 - Simpler models may be preferred for robustness and debugging ease.
- 7. How does model performance differ in A/B testing compared to offline validation?
 - A/B testing reflects live user behavior, real data drift.
 - Offline scores are static they may not capture live system changes, feedback loops, or edge cases.

NLP (Tricky and Subtle Questions)

- 1. You trained a BERT model for sentiment analysis, but it fails on slang-heavy texts what's happening?
 - BERT wasn't fine-tuned on domain-specific or informal language.
 - Slang may be OOV (out of vocabulary) or poorly tokenized.
- 2. Can tokenization errors affect downstream model accuracy? How?
 - Yes incorrect token boundaries misrepresent word meaning/context.
 - Especially crucial in subword models where part of the semantic structure can be lost.
- 3. Why might TF-IDF perform better than contextual embeddings in a simple text classification task?
 - TF-IDF captures term frequency well for short, topic-specific tasks.
 - BERT might overfit on small datasets or add unnecessary complexity.
- 4. If two prompts give drastically different outputs from an LLM, what could be the cause?
 - LLMs are highly sensitive to phrasing, structure, and context length.
 - Prompting acts as soft conditioning small changes may shift the output distribution.

5. What's the limitation of using BLEU or ROUGE as evaluation metrics for generative models?

- They rely on surface-level word overlap, not meaning.
- o Can't capture paraphrases or semantically correct outputs with different phrasing.

✓ Statistics & Probability Questions

1. Can a statistically significant result be practically useless? Explain with an example.

- Yes a drug reduces blood pressure by 0.5 mmHg with p < 0.001. Statistically significant, but not medically useful.
- 2. Why is p-value misunderstood in most real-world use cases?
 - o It's often misinterpreted as the probability that the null is true.
 - o Actually, it's the probability of observing the data assuming the null is true.
- 3. In what cases can increasing sample size worsen your model?
 - More noisy or irrelevant data can confuse the model.
 - Computational cost and overfitting to noise might increase.
- 4. If confidence intervals of two models overlap, can you claim one is better?
 - Not conclusively overlapping CIs suggest uncertainty.
 - Perform hypothesis testing or use bootstrapping to compare.
- 5. Why is correlation not enough to infer causation, even if the p-value is low?
 - o Hidden confounders may affect both variables.
 - Correlation is symmetric causation is directional.

Classification Edge Cases

- 1. Your model predicts only one class in an imbalanced dataset even after SMOTE. What next?
 - Use different resampling methods (e.g., ADASYN).
 - o Try cost-sensitive learning or focal loss.
 - Re-check your preprocessing and model capacity.
- 2. A model shows 90% accuracy but misses most minority class instances what's the fix?
 - Accuracy is misleading focus on recall, F1-score for minority class.
 - Rebalance data, tune threshold, or use ensemble models.
- 3. What is the impact of threshold tuning on precision and recall trade-off?
 - Lowering threshold increases recall but decreases precision.
 - Tuning helps optimize for business-specific trade-offs.
- 4. When would you choose F1-score over ROC-AUC, and vice versa?
 - F1-score: imbalanced classes and when false negatives matter.
 - o ROC-AUC: when you care about ranking or probability calibration.
- 5. You get great metrics on test data but poor results in live predictions what could be wrong?
 - Data drift or concept drift.
 - Training-serving skew or pipeline bugs.

Test data not representative of real-world input.

Machine Learning Questions (Tricky but Basic)

1. Can dropping features improve model performance? How do you decide?

- Yes, by removing irrelevant or noisy features.
- Use feature importance, correlation analysis, or recursive feature elimination.

2. Why is data imbalance not always a problem?

- If the cost of false negatives/positives is low or the task is ranking, imbalance may not affect much.
- Some models can still learn patterns well with imbalance.

3. What's the difference between randomness and noise in training data?

- Randomness is inherent variability, noise is unwanted distortion (e.g., mislabeled data).
- Noise reduces learnability.

4. Why might a model's performance decrease after hyperparameter tuning?

- Overfitting to validation set.
- Poor search space or incorrect objective.

5. How does imputation of missing values introduce bias?

- o Imputation may assume patterns that don't exist.
- Mean/median imputation can distort distributions.

6. When can scaling after train-test split lead to data leakage?

- If scalers are fit on entire dataset before splitting.
- Statistics from test set leak into training.

7. What's the risk of using polynomial features blindly in linear models?

Overfitting, increased dimensionality, instability.

8. Can high F1-score be misleading? When?

• Yes, when classes are imbalanced and F1 hides poor precision or recall.

9. Why is model interpretability important in regulated industries?

- o To ensure fairness, transparency, compliance.
- Legal or ethical reasons.

10. Why should you be cautious about reusing test data for model selection?

- Leads to test set leakage.
- Makes the test set no longer a valid measure of generalization.

✓ Natural Language Processing (NLP) Questions

1. How does domain-specific vocabulary affect pre-trained NLP models?

- Poor tokenization or missing embeddings.
- Reduced accuracy due to OOV or shifted semantics.

2. What is the risk of using sentiment lexicons across different domains (e.g., finance vs. movies)?

Words have different sentiment meanings in different contexts.

- Domain adaptation is needed.
- 3. Why does length normalization matter in text similarity tasks?
 - Longer texts may get unfairly higher scores.
 - Normalization ensures fair comparisons.
- 4. What's the issue with blindly using TF-IDF in very short texts?
 - Sparse vectors, unstable weights.
 - May miss context or semantics.
- 5. Why might a generative model repeat phrases in its outputs?
 - Exposure bias, poor sampling.
 - Weak penalties for repetition.
- 6. How can POS tagging errors cascade into bigger NLP mistakes?
 - o Downstream tasks (e.g., parsing, entity recognition) rely on POS tags.
 - Errors propagate and affect predictions.

Statistics & Probability Questions (Advanced)

- 1. Why is standard deviation not always the best measure of spread?
 - Sensitive to outliers.
 - Doesn't reflect skewed distributions well.
- 2. What's the subtle trap in interpreting p-values near 0.05?
 - Arbitrary threshold may lead to false significance claims.
 - o Results near 0.05 should be treated with caution.
- 3. Can two distributions have the same mean and variance but still be very different?
 - Yes different shapes (e.g., bimodal vs. unimodal).
- 4. Why is random sampling not truly random in code?
 - Uses pseudo-random generators.
 - May repeat if seed is fixed.
- 5. What is the danger in aggregating data without stratification?
 - Simpson's paradox.
 - Bias due to hidden group effects.

✓ Classification Questions (Advanced)

- 1. How does label noise affect classifier training?
 - Model learns wrong patterns.
 - Reduced accuracy and generalization.
- 2. Why can a high AUC still mean a poor model for threshold-based decisions?
 - AUC considers ranking, not actual thresholded predictions.
 - Poor calibration can still yield high AUC.
- 3. When is log-loss a better metric than accuracy?
 - When probability confidence matters.
 - o In imbalanced datasets.
- 4. How can overly confident predictions hurt in real-world applications?
 - Misleads decision-makers.
 - Amplifies errors when wrong.

- 5. Why is multiclass classification harder than it seems with one-vs-rest?
 - Overlap between classes.
 - Class imbalance and inconsistent thresholds.

Machine Learning Questions (Tricky but Basic - Part 2)

- 1. Why might a model perform worse on real-world data even after achieving great cross-validation results?
 - Cross-validation may not capture real-world distribution shifts.
 - o Data drift, sampling bias, or leakage in real usage can degrade performance.
- 2. How does feature scaling impact gradient descent performance?
 - It improves convergence speed.
 - Prevents oscillation by ensuring features contribute equally to updates.
- 3. Why can adding polynomial features lead to overfitting in linear models?
 - Increases dimensionality and model flexibility.
 - Captures noise rather than signal on limited data.
- 4. How is entropy used in decision trees, and what does it really represent?
 - Entropy quantifies impurity or disorder in the data.
 - Decision trees use it to choose splits that reduce uncertainty.
- 5. Why doesn't increasing model complexity always improve accuracy?
 - Higher complexity increases risk of overfitting.
 - May fit noise and reduce generalization.
- 6. What's the intuition behind k-nearest neighbors being a "lazy learner"?
 - No learning during training; stores data and defers computation to inference.
 - Makes predictions by comparing to stored examples.
- 7. How do learning rate and batch size interact in training a deep learning model?
 - Small batch size allows faster convergence but with noisy updates.
 - Larger batch size requires smaller learning rates to avoid divergence.
- 8. Why are tree-based models invariant to feature scaling, but linear models are not?
 - o Tree splits depend on thresholds, not magnitudes.
 - Linear models rely on weight updates sensitive to feature magnitude.
- 9. In boosting algorithms, how does the model correct previous errors?
 - Each new learner focuses more on misclassified instances from prior models.
 - Adjusts weights or residuals to reduce cumulative error.
- 10. Why is interpretability often a trade-off with model accuracy?
- Complex models (e.g., ensembles, deep nets) are hard to interpret.
- Simple models may be transparent but less expressive.

✓ Natural Language Processing (NLP) Questions - Part 2

- 1. Why is tokenization a non-trivial step in NLP pipelines?
 - Token definitions vary by language and application.
 - Poor tokenization affects downstream embedding and learning.
- 2. How do attention mechanisms differ from traditional RNNs in handling long sequences?

- Attention looks at all tokens simultaneously.
- RNNs process sequentially and suffer from vanishing gradients.

3. Why can padding and truncation impact model performance in NLP?

- Padding introduces artificial tokens.
- Truncation may remove important context.

4. What are stopwords, and when should you avoid removing them?

- Common words (e.g., "is", "not").
- o In sentiment tasks, words like "not" can flip meaning should be retained.

5. Why are word embeddings better than one-hot encoding?

- o Embeddings capture semantic similarity.
- One-hot vectors are sparse and lack context.

6. How do named entity recognition systems deal with ambiguous entities?

- Use context and language models.
- May rely on CRFs or BERT-based contextual cues.

Statistics & Probability Questions - Part 2

1. What's the law of large numbers, and how does it differ from the central limit theorem?

- LLN: sample mean converges to population mean as $n \rightarrow \infty$.
- CLT: distribution of sample means becomes normal regardless of population shape.

2. Why can two datasets have the same mean but different standard deviations?

- Spread or dispersion of values differs.
- One dataset may have outliers or wider range.

3. How does correlation differ from causation, and why is this distinction critical?

- Correlation = association; causation = one causes the other.
- Acting on correlation without causality can mislead decisions.

4. Why is standard error smaller than standard deviation in most cases?

- \circ SE = SD / sqrt(n).
- Aggregating samples reduces variability.

5. What are p-values often misunderstood to represent?

- Misread as probability that null is true.
- Actually, probability of observed data under null hypothesis.

Classification Questions - Part 2

1. Why can accuracy be misleading in imbalanced datasets?

- High accuracy possible by predicting only majority class.
- Ignores poor performance on minority class.

2. What is precision-recall trade-off, and how does it affect model choice?

- Precision: correctness of positives.
- Recall: coverage of actual positives.
- Choose based on false positive vs. false negative cost.

3. How does threshold tuning impact classification metrics?

- Alters balance of precision and recall.
- Helps optimize for task-specific needs.

4. Why might you prefer F1-score over accuracy in medical diagnoses?

- Medical tasks are imbalanced and sensitive to false negatives.
- F1 balances both precision and recall.

5. What does it mean if a classifier has high precision but low recall?

- o It predicts positives with confidence but misses many actual positives.
- May be too conservative needs threshold adjustment.

Machine Learning Questions (Tricky but Basic - Part 3)

1. Why can a high training accuracy be misleading when evaluating a model's performance?

- It may indicate overfitting to training data.
- High training accuracy doesn't guarantee generalization to unseen data.

2. What's the intuition behind the bias-variance trade-off?

- Bias: error from overly simple models.
- Variance: error from overly complex models.
- Goal is to find a balance where total error is minimized.

3. Why is random initialization crucial in training neural networks?

- o Prevents symmetry where neurons learn the same features.
- Helps explore different regions of the loss surface.

4. In ensemble methods like random forests, why is averaging predictions better than taking a majority vote?

- Averaging reduces variance and produces smoother outputs.
- Especially useful in regression or probability outputs.

5. How does L1 regularization differ from L2, and when would you use each?

- L1 leads to sparse models (feature selection).
- L2 penalizes large weights and keeps all features.
- Use L1 when feature reduction is desired.

6. What's the purpose of using an activation function in neural networks?

- Introduces non-linearity.
- Allows networks to approximate complex functions.

7. How does increasing the number of estimators in a random forest impact model performance?

- Reduces variance and overfitting.
- Eventually reaches diminishing returns and adds computational cost.

8. Why can using too many features lead to worse model performance, and how can you detect it?

- Adds noise, increases overfitting risk.
- Use feature importance or regularization techniques to detect.

9. How does early stopping work in deep learning, and why is it useful?

- Stops training when validation loss stops improving.
- Prevents overfitting and saves computation.
- 10. What is data leakage, and how can it affect your machine learning model?

- When information from the target leaks into training data.
- Leads to overestimated performance and unreliable predictions.

Natural Language Processing (NLP) Questions - Part 3

- 1. Why is stemming sometimes less accurate than lemmatization?
 - Stemming crudely chops words, may distort meaning.
 - Lemmatization uses grammar rules and vocabulary for accurate base forms.
- 2. How does positional encoding help transformer-based models understand word order?
 - Injects order information into embeddings.
 - Allows attention mechanism to differentiate token positions.
- 3. What are the challenges in handling code-mixed languages (e.g., Hinglish) in NLP?
 - Inconsistent grammar, spelling, and switching contexts.
 - Lack of annotated datasets and standardized tokenization.
- 4. Why are subword tokenization techniques (like Byte-Pair Encoding) preferred in large language models?
 - Handles rare and out-of-vocabulary words.
 - Reduces vocabulary size while preserving meaning.
- 5. What is the vanishing gradient problem, and how do RNNs suffer from it?
 - o Gradients shrink during backpropagation.
 - o Prevents learning long-term dependencies.
- 6. Why is fine-tuning pre-trained models more effective than training from scratch in NLP?
 - Pre-trained models already capture language structure.
 - Fine-tuning adapts them to specific tasks with less data.

✓ Statistics & Probability Questions - Part 3

- 1. What is the central limit theorem, and why is it important in data science?
 - Sample means follow a normal distribution even if population isn't.
 - Enables use of inferential statistics on arbitrary distributions.
- 2. When would you prefer a non-parametric test over a parametric one?
 - When data doesn't meet normality or variance assumptions.
 - Use for ordinal data or small sample sizes.
- 3. Why does increasing sample size reduce sampling error but not necessarily eliminate bias?
 - Sampling error = variability due to randomness.
 - Bias comes from flawed design or assumptions not fixed by sample size.
- 4. How do confidence intervals help interpret statistical results?
 - Provide a range for parameter estimates.
 - Express uncertainty and reliability of estimates.
- 5. What is the difference between Type I and Type II errors, and how do they affect model decisions?

- Type I: false positive rejecting true null.
- Type II: false negative failing to reject false null.
- Balance affects sensitivity vs. specificity.

✓ Classification Questions - Part 3

- 1. What is a confusion matrix, and how does it help evaluate model performance?
 - Shows true vs. predicted labels.
 - Reveals distribution of false positives/negatives.
- 2. Why is ROC-AUC a better evaluation metric in imbalanced classification problems?
 - Measures true positive rate vs. false positive rate.
 - o Considers all thresholds robust to imbalance.
- 3. How can class imbalance be addressed during training?
 - Use resampling techniques (SMOTE, undersampling).
 - Try class weights or specialized loss functions.
- 4. What's the role of regularization in logistic regression?
 - Penalizes large weights to reduce overfitting.
 - o Improves generalization.
- 5. How does a decision boundary change when using different kernels in SVM?
 - o Linear kernel creates a straight boundary.
 - o RBF/poly kernels create curved, complex boundaries.
 - Kernel choice affects flexibility and fit.